WHAT IS CLAIMED IS:

1. A method of encompressing a data stream, comprising:

compressing vectors from the data stream using one or more Multiple Attractor Cellular Automatas (MACAs); and

encrypting the compressed vectors using multiple Cellular Automata (CA) transforms.

2. The method of Claim 1, wherein compressing the vectors and encrypting the compressed vectors is a single integrated process implemented with a program executed on a Programmable CA (PCA).

3. The method of Claim 1, further comprising generating a code-book, the one or more MACAs operable to perform binary searches in the code-book to compress the vectors from the data stream.

4. The method of Claim 3, further comprising storing the code-book using one or more multi-stage MACA-based two class classifiers which act as implicit memory to store the code-book.

5. The method of Claim 1, wherein compressing the vectors from the data stream using one or more MACAs comprises deriving code-book indices for the vectors.

6. The method of Claim 1, wherein encrypting the compressed vectors using multiple CA transforms comprises using a series of reversible transforms that use one or more of linear CA, additive CA, and non-linear CA configured in a PCA at one or more different time steps.

7. The method of Claim 6, comprising encrypting the compressed vectors using four levels of CA transforms.

8. The method of Claim 7, wherein encrypting the compressed vectors using multiple CA transforms comprises using one or more of linear transformations, affine transformations, and non-affine transformations.

9. The method of Claim 1, further comprising transmitting the encompressed data across a communications link.

10. The method of Claim 9, further comprising decrypting the transmitted encompressed data using multiple CA transforms.

11. Logic encoded in media for encompressing a data stream, when executed the logic operable to:

compress vectors from the data stream using one or more Multiple Attractor Cellular Automatas (MACAs); and

encrypt the compressed vectors using multiple Cellular Automata (CA) transforms.

12. The logic of Claim 11, operable to compress the vectors and encrypt the compressed vectors in a single integrated process implemented with a Programmable CA (PCA).

13. The logic of Claim 11, further operable to generate a code-book, the one or more MACAs operable to perform binary searches in the code-book to compress the vectors from the data stream.

14. The logic of Claim 11, further operable to store the code-book using one or more multi-stage MACA-based two class classifiers which act as implicit memory to store the code-book.

15. The logic of Claim 11, operable to compress the vectors from the data stream using one or more MACAs by deriving code-book indices for the vectors.

16. The logic of Claim 11, operable to encrypt the compressed vectors using multiple CA transforms by using a series of reversible transforms that use one or more of linear CA, additive CA, and non-linear CA configured in a PCA at one or more different time steps.

17. The logic of Claim 15, operable to encrypting the compressed vectors using four levels of CA transforms.

18. The logic of Claim 17, operable to encrypt the compressed vectors using multiple CA transforms by using one or more of linear transformations, affine transformations, and non-affine transformations.

19. The logic of Claim 11, further operable to transmit the encompressed data across a communications link.

20. The logic of Claim 19, further operable to decrypt the transmitted encompressed data using multiple CA transforms.

21. A system for encompressing a data stream, the system comprising:

a first module operable to compress vectors from the data stream using one or more Multiple Attractor Cellular Automatas (MACAs); and

a second module operable to encrypt the compressed vectors using multiple Cellular Automata (CA) transforms.

22. The system of Claim 21, wherein the first module is operable to compress the vectors and the second module is operable to encrypt the compressed vectors in a single integrated process implemented with a Programmable CA (PCA).

23. The system of Claim 21, wherein the first module is further operable to generate a code-book, the one or more MACAs operable to perform binary searches in the code-book to compress the vectors from the data stream.

24. The system of Claim 21, wherein the first module is further operable to store the code-book using one or more multi-stage MACA-based two class classifiers which act as implicit memory to store the code-book.

25. The system of Claim 21, wherein the first module is operable to compress the vectors from the data stream using one or more MACAs by deriving code-book indices for the vectors.

26. The system of Claim 21, wherein the second module is operable to encrypt the compressed vectors using multiple CA transforms by using a series of reversible transforms that use one or more of linear CA, additive CA, and non-linear CA configured in a PCA at one or more different time steps.

27. The system of Claim 26, wherein the second module is operable to encrypt the compressed vectors using four levels of CA transforms.

28. The system of Claim 27, wherein the second module is operable to encrypt the compressed vectors using multiple CA transforms by using one or more of linear transformations, affine transformations, and non-affine transformations.

29. The system of Claim 21, further operable to transmit the encompressed data across a communications link.

30. The system of Claim 27, further operable to decrypt the transmitted encompressed data using multiple CA transforms.

31. A system for encompressing a data stream, comprising:

a programmable CA (PCA) operable to receive vectors from the data stream;

a program memory operable to communicate with the PCA;

an index memory operable to communicate with the PCA; and

an index register operable to communicate with the index memory;

the program memory storing a program operable to:

configure the PCA with a rule vector of a CA; and

enable the PCA to be run through a number of cycles controlled by the program, a resulting Pseudo-Exhaustive Field (PEF) value being directed to address the index memory;

the index memory providing values to the index register, enabling a code-book index to be generated for an input vector loaded into the PCA.

32. The system of Claim 31, wherein the index memory comprises a 1-bit index memory, the resulting value directed to the index memory comprising a PEF of an attractor state of the CA.

33. A method for encompressing a data stream, comprising:

receiving at a programmable CA (PCA) vectors from the data stream;

using a program stored in a program memory in communication with the PCA:

configuring the PCA with a rule vector of a CA; and

enabling the PCA to be run through a number of cycles controlled by the program, a resulting Pseudo-Exhaustive Field (PEF) value being directed to address an index memory in communication with the PCA, the index memory providing values to an index register in communication with the index memory, enabling a code-book index to be generated for an input vector loaded into the PCA.

34. The method of Claim 33, wherein:

the index memory comprises a 1-bit index memory; and

the resulting value comprises a PEF of an attractor state of the CA.